# Machine Learning Basics
# Lecture 5: SVM II

Princeton University COS 495

Instructor: Yingyu Liang

# Review: SVM objective

# SVM: objective

- Let $y_i \in \{+1, -1\}$, $f_{w,b}(x) = w^T x + b$. Margin:

$$\gamma = \min_i \frac{y_i f_{w,b}(x_i)}{||w||}$$

- Support Vector Machine:

$$\max_{w,b} \gamma = \max_{w,b} \min_i \frac{y_i f_{w,b}(x_i)}{||w||}$$

# SVM: optimization

- Optimization (Quadratic Programming):

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$y_i(w^T x_i + b) \geq 1, \forall i$$

- Solved by Lagrange multiplier method:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \frac{1}{2}||w||^2 - \sum_i \alpha_i [y_i(w^T x_i + b) - 1]$$

where $\boldsymbol{\alpha}$ is the Lagrange multiplier

# Lagrange multiplier

# Lagrangian

- Consider optimization problem:

$$\min_{w} f(w)$$

$$h_i(w) = 0, \forall 1 \leq i \leq l$$

- Lagrangian:

$$\mathcal{L}(w, \boldsymbol{\beta}) = f(w) + \sum_i \beta_i h_i(w)$$

where $\beta_i$'s are called Lagrange multipliers

# Lagrangian

- Consider optimization problem:

$$\min_{w} \ f(w)$$

$$h_i(w) = 0, \forall 1 \leq i \leq l$$

- Solved by setting derivatives of Lagrangian to $0$

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0; \quad \frac{\partial \mathcal{L}}{\partial \beta_i} = 0$$

# Generalized Lagrangian

- Consider optimization problem:

$$\min_{w} f(w)$$

$$g_i(w) \leq 0, \forall 1 \leq i \leq k$$

$$h_j(w) = 0, \forall 1 \leq j \leq l$$

- Generalized Lagrangian:

$$\mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(w) + \sum_i \alpha_i g_i(w) + \sum_j \beta_j h_j(w)$$

where $\alpha_i, \beta_j$'s are called Lagrange multipliers

# Generalized Lagrangian

- Consider the quantity:

$$\theta_P(w) := \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- Why?

$$\theta_P(w) = \begin{cases} f(w), & \text{if } w \text{ satisfies all the constraints} \\ +\infty, & \text{if } w \text{ does not satisfy the constraints} \end{cases}$$

- So minimizing $f(w)$ is the same as minimizing $\theta_P(w)$

$$\min_w f(w) = \min_w \theta_P(w) = \min_w \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

# Lagrange duality

- The primal problem

$$p^* := \min_w f(w) = \min_w \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- The dual problem

$$d^* := \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \min_w \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- Always true:

$$d^* \leq p^*$$

# Lagrange duality

- The primal problem

$$p^* := \min_w f(w) = \min_w \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- The dual problem

$$d^* := \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \min_w \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- Interesting case: when do we have

$$d^* = p^*?$$

# Lagrange duality

- Theorem: under proper conditions, there exists $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ such that

$$d^* = \mathcal{L}(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = p^*$$

Moreover, $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ satisfy Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0, \qquad \alpha_i g_i(w) = 0$$

$$g_i(w) \leq 0, \;\; h_j(w) = 0, \qquad \alpha_i \geq 0$$

# Lagrange duality

- Theorem: under proper conditions, there exists $(w$

dual complementarity

$$d^* = \mathcal{L}(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = p^*$$

Moreover, $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ satisfy Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0, \qquad \alpha_i g_i(w) = 0$$

$$g_i(w) \le 0, \;\; h_j(w) = 0, \qquad \alpha_i \ge 0$$

# Lagrange duality

primal constraints

conditions, there exists $(w$

dual constraints

$$= \mathcal{L}(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = p^*$$

- Moreover, $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ satisfy Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0, \qquad \alpha_i g_i(w) = 0$$

$$g_i(w) \leq 0, \; h_j(w) = 0, \qquad \alpha_i \geq 0$$

# Lagrange duality

- What are the proper conditions?
- A set of conditions (Slater conditions):
  - $f, g_i$ convex, $h_j$ affine
  - Exists $w$ satisfying all $g_i(w) < 0$

- There exist other sets of conditions
  - Search Karush–Kuhn–Tucker conditions on Wikipedia

# SVM: optimization

# SVM: optimization

- Optimization (Quadratic Programming):

$$\min_{w,b} \frac{1}{2}\left|\left|w\right|\right|^2$$

$$y_i(w^T x_i + b) \geq 1, \forall i$$

- Generalized Lagrangian:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \frac{1}{2}\left|\left|w\right|\right|^2 - \sum_i \alpha_i [y_i(w^T x_i + b) - 1]$$

where $\boldsymbol{\alpha}$ is the Lagrange multiplier

# SVM: optimization

- KKT conditions:

$$\frac{\partial \mathcal{L}}{\partial w} = 0, \rightarrow w = \sum_i \alpha_i y_i x_i \quad (1)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0, \rightarrow 0 = \sum_i \alpha_i y_i \quad (2)$$

- Plug into $\mathcal{L}$:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (3)$$

combined with $0 = \sum_i \alpha_i y_i$, $\alpha_i \geq 0$

# SVM: optimization

Only depend on inner products

- Reduces to dual problem:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\sum_i \alpha_i y_i = 0, \alpha_i \geq 0$$

- Since $w = \sum_i \alpha_i y_i x_i$, we have $w^T x + b = \sum_i \alpha_i y_i x_i^T x + b$

# Kernel methods

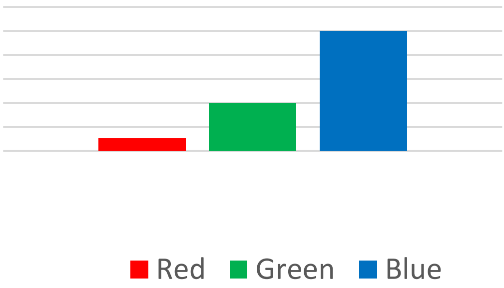
# Features

$x$

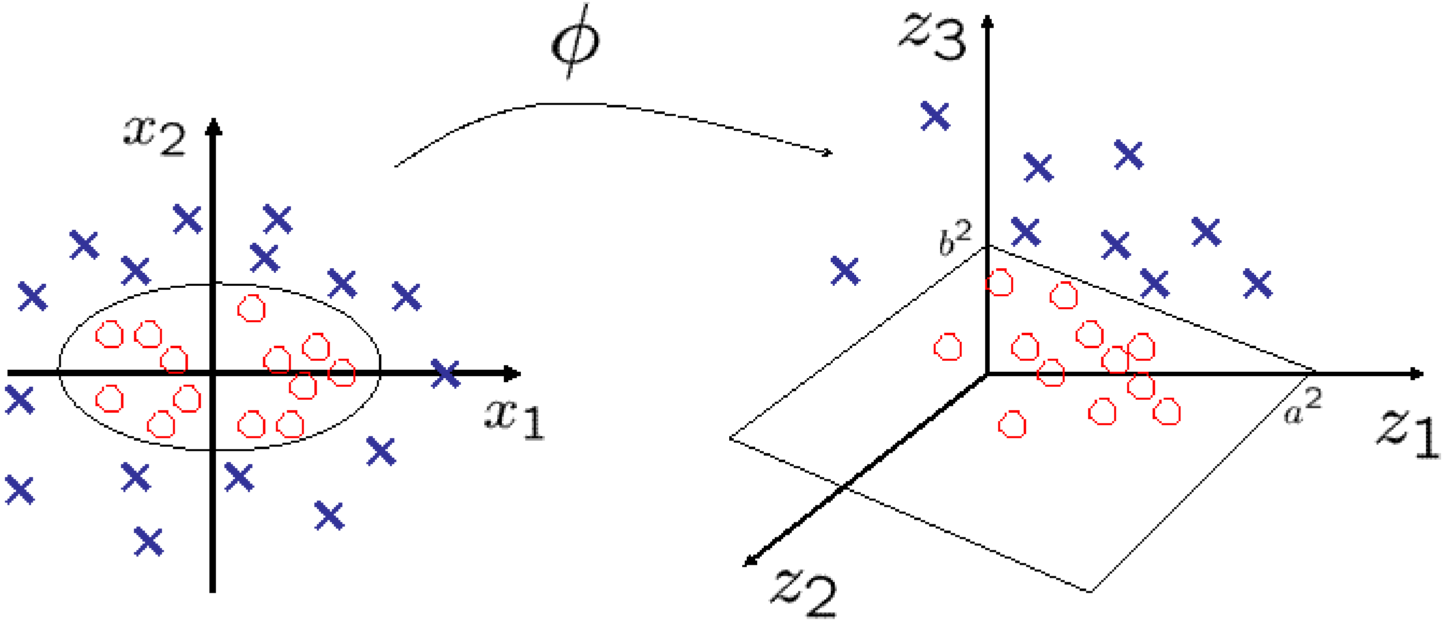
Extract features

$\phi(x)$

Color Histogram



# Features




$$\phi : (x_1, x_2) \longrightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

$$\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 = 1 \longrightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = 1$$

# Features

- Proper feature mapping can make non-linear to linear
- Using SVM on the feature space $\{\phi(x_i)\}$: only need $\phi(x_i)^T\phi(x_j)$

- Conclusion: no need to design $\phi(\cdot)$, only need to design

$$k(x_i, x_j) = \phi(x_i)^T\phi(x_j)$$

# Polynomial kernels

- Fix degree $d$ and constant $c$:

$$k(x, x') = (x^T x' + c)^d$$

- What are $\phi(x)$?
- Expand the expression to get $\phi(x)$

# Polynomial kernels

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^2, \quad K(\mathbf{x}, \mathbf{x}') = (x_1 x_1' + x_2 x_2' + c)^2 = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} {x'}_1^2 \\ {x'}_2^2 \\ \sqrt{2}\, x_1' x_2' \\ \sqrt{2c}\, x_1' \\ \sqrt{2c}\, x_2' \\ c \end{bmatrix}$$

Figure from Foundations of Machine Learning, by M. Mohri, A. Rostamizadeh, and A. Talwalkar

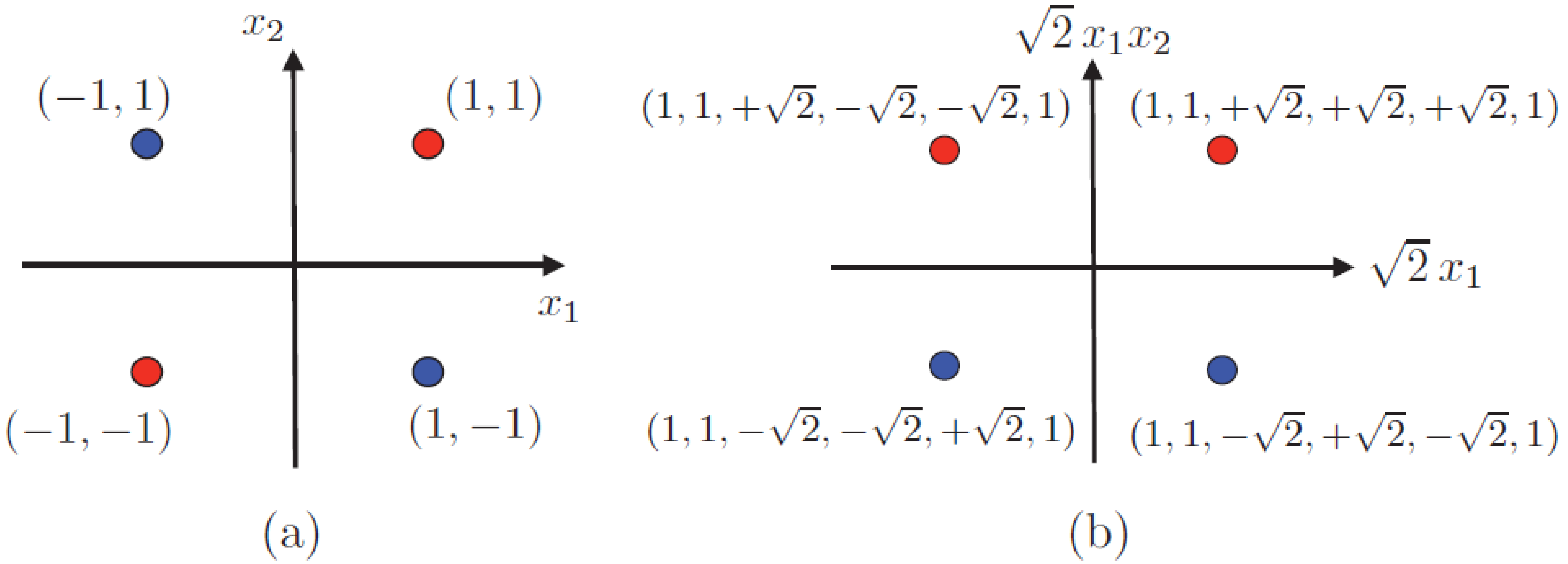



**Figure 5.2** Illustration of the XOR classification problem and the use of polynomial kernels. (a) XOR problem linearly non-separable in the input space. (b) Linearly separable using second-degree polynomial kernel.

Figure from Foundations of Machine Learning, by M. Mohri, A. Rostamizadeh, and A. Talwalkar

# Gaussian kernels

- Fix bandwidth $\sigma$:

$$k(x, x') = \exp(-\left||x - x'|\right|^2 / 2\sigma^2)$$

- Also called radial basis function (RBF) kernels

- What are $\phi(x)$? Consider the un-normalized version

$$k'(x, x') = \exp(x^T x' / \sigma^2)$$

- Power series expansion:

$$k'(x, x') = \sum_{i}^{+\infty} \frac{(x^T x')^i}{\sigma^i i!}$$

# Mercer's condition for kenerls

- Theorem: $k(x, x')$ has expansion

$$k(x, x') = \sum_{i}^{+\infty} a_i \phi_i(x) \phi_i(x')$$

if and only if for any function $c(x)$,

$$\int\int c(x)c(x')k(x, x')dxdx' \geq 0$$

(Omit some math conditions for $k$ and $c$)

# Constructing new kernels

- Kernels are closed under positive scaling, sum, product, pointwise limit, and composition with a power series $\sum_i^{+\infty} a_i k^i(x, x')$

- Example: $k_1(x, x'), k_2(x, x')$ are kernels, then also is

$$k(x, x') = 2k_1(x, x') + 3k_2(x, x')$$

- Example: $k_1(x, x')$ is kernel, then also is

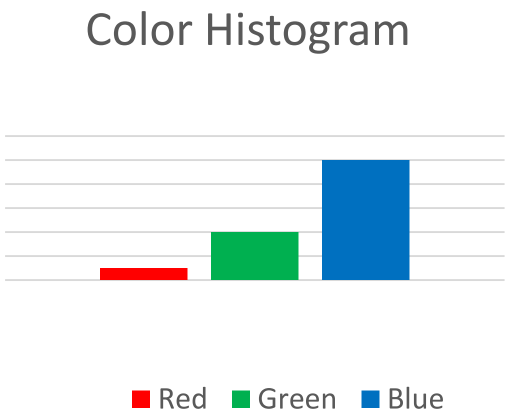
$$k(x, x') = \exp(k_1(x, x'))$$

# Kernels v.s. Neural networks

# Features

$x$

Extract features

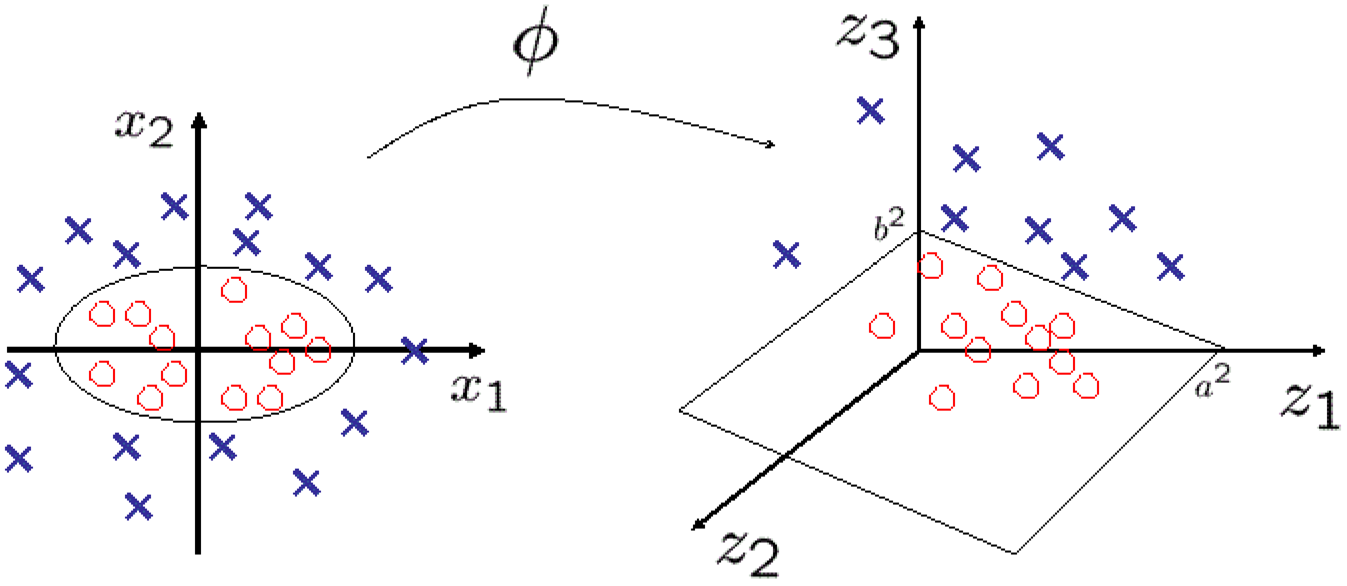



build hypothesis

$$y = w^T \phi(x)$$

# Features: part of the model

Nonlinear model




build hypothesis

$y = w^T \phi(x)$

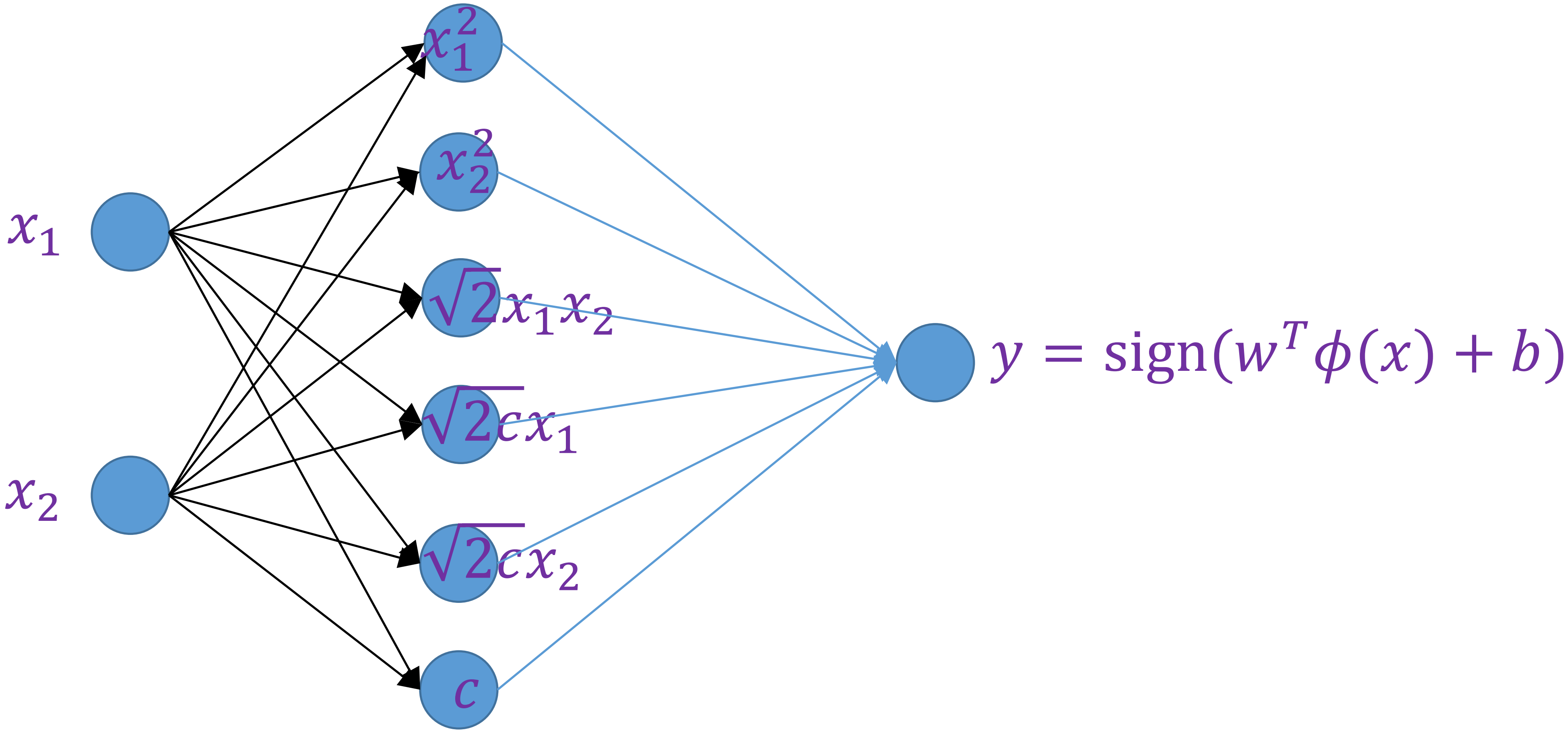
Linear model

# Polynomial kernels

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^2, \quad K(\mathbf{x}, \mathbf{x}') = (x_1 x_1' + x_2 x_2' + c)^2 = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} {x'}_1^2 \\ {x'}_2^2 \\ \sqrt{2}\, x_1' x_2' \\ \sqrt{2c}\, x_1' \\ \sqrt{2c}\, x_2' \\ c \end{bmatrix}$$

Figure from Foundations of Machine Learning, by M. Mohri, A. Rostamizadeh, and A. Talwalkar

# Polynomial kernel SVM as two layer neural network




First layer is fixed. If also learn first layer, it becomes two layer neural network